ArXiv manuscript

Non-parametric likelihood based estimation of linear filters for point processes

Niels Richard Hansen

Abstract We consider models for multivariate point processes where the intensity is given non-parametrically in terms of functions in a reproducing kernel Hilbert space. The likelihood function involves a time integral and is consequently not given in terms of a finite number of kernel evaluations. We derive a representation of the gradient of the log-likelihood and provide two methods for practically computing approximations to the gradient by time discretization. We illustrate the methods by an application to neuron network modeling, and we investigate how the computational costs of the methods depend on the resolution of the time discretization. The methods are implemented and available in the R-package `ppstat`.

Keywords Multivariate point processes · Penalization · Reproducing kernel Hilbert spaces

1 Introduction

A network of neurons is a prime example of an interacting dynamical system, and the characterization and modeling of the network activity is a central scientific challenge, see e.g. [11]. In this paper we consider models and related computational methods appropriate for the modeling of the collection of spike times, which can be measured simultaneously for multiple neurons. The spike times are discrete event times and the appropriate modeling framework is that of multivariate counting or point process models. Our results are general, but their application is illustrated on neuron spike data.

Niels Richard Hansen
University of Copenhagen, Department of Mathematical Sciences, Universitetsparken 5, 2100 Copenhagen, Denmark.
Tel.: +45 - 35 32 07 83
E-mail: Niels.R.Hansen@math.ku.dk

With (N_t^i) denoting the counting process of spike times for neuron i, $i = 1, \ldots, p$, a classical linear Hawkes model, [9], is specified by letting the intensity of a spike for the i'th neuron be given as

$$X_t^i = \sum_{j=1}^{p} \int_0^{t-} e^{\alpha_{ij}(t-s)+\beta_{ij}} \, \mathrm{d}N_s^j. \tag{1}$$

This intensity, or rate, specifies the conditional probability of observing a spike immediately after time t in the sense that

$$P(N_{t+\delta}^i - N_t^i = 1 \mid \mathcal{F}_t) \simeq \delta X_t^i$$

where $\mathcal{F}_t$ denotes the history of all spikes preceeding time t, see e.g. [10] or [1]. Note the upper integration limit, $t-$, which means that the integral w.r.t. N_s^j only involves spikes strictly before t. This is an essential requirement for correct likelihood computations, see (3) below.

Neuron network activity is one example of a multivariate interacting dynamical system that is driven by discrete events, but there are also other important examples. For instance the high-frequency trading of multiple financial assets [8]. Indeed, the linear Hawkes model (1) was also considered in Chapter 7 in [8]. Other examples are found in epidemiology, population biology and (bio)chemistry. Examples of chemical reaction networks using Markovian multitype birth-death processes are found in [2] and [4]. For such models of chemical reactions we introduce a vector of *state variables* consisting of the numbers of molecules of each type of reactant at a given time. The intensities of reaction events are then expressed in terms of the state variables and a collection of rate parameters. For the model given by (1) we can also introduce $(X_t^1, \ldots, X_t^p)$ as a vector of state variables. This p-dimensional process is then a Markov

process and there is a one-to-one correspondance between this process and the multivariate counting process $(N_t^1, \ldots, N_t^p)$. In the special case $\alpha_{ij} = 0$ the model reduces to a linear birth process. In applications we are, however, more interested in cases where $\alpha_{ij} < 0$ so that the state variables constitute cumulated but exponentially damped contributions of previous events. The Markovian birth-death models and the model given by (1) share two things; the state variables form a continuous time Markov process, and the models are specified by a finite dimensional parameter vector. The Markov property of the state variables is, among other things, beneficial from a computational viewpoint, and standard techniques can be used for likelihood computations and parameter estimation given that we observe the complete system.

Our interest is to generalize the model given by (1) to non-exponential integrands, and, in particular, to allow those integrands to be estimated non-parametrically. A consequence is that the Markov property of the state variables will be lost. The integral (1) can be understood as a linear filter of the multivariate counting process $(N_t^1, \ldots, N_t^p)$, and we will consider the generalization of such linear filters to the case where

$$X_t^i = \sum_{j=1}^p \int_0^t g_{ij}(t-s) \, \mathrm{d}N_s^j \tag{2}$$

with g_{ij} general functions in a suitable function space. We will, moreover, allow for non-linear transformations of X_t^i, such that the intensity is given by $\varphi(X_t^i)$ for a general but fixed function φ.

In this paper we are particularly concerned with efficient computation and minimization of the penalized negative log-likelihood as a function of the non-parametric components g_{ij}, with g_{ij} in a reproducing kernel Hilbert space $\mathcal{H}$. We consider algorithms for standard quadratic penalization $\sum_{ij} ||g_{ij}||^2$, with $||\cdot||$ the Hilbert space norm on $\mathcal{H}$. We will throughout assume that the g_{ij}-functions are variation independent, which imply that the computation and minimization of the joint penalized negative log-likelihood can be split into p separate minimization problems. To ease notation we will thus subsequently consider the modeling of one counting process N_t in terms of $N_t^1, \ldots, N_t^p$, where N_t can be any of the p counting processes.

2 Likelihood computations for point processes specified by linear filters

We assume that we observe a simple counting process $(N_s)_{0 \leq s \leq t}$ of discrete events on the time interval $[0, t]$. The jump times of N are denoted $\tau_1 < \ldots < \tau_{N_t}$. We let $\mathcal{H}$ denote a reproducing kernel Hilbert space of functions on $[0, t]$ with reproducing kernel $R : [0, t] \times [0, t] \to \mathbb{R}$, and we let $g = (g_1, \ldots, g_p) \in \mathcal{H}^p$. We assume that R is continuous in which case the functions in $\mathcal{H}$ are also continuous, see Theorem 17 in [3]. With $N^1, \ldots, N^p$ counting processes having jump times σ_j^i we introduce

$$X_s(g) = \sum_{i=1}^p \int_0^{s-} g_i(s-u) \, \mathrm{d}N_u^i = \sum_{i=1}^p \sum_{j : \sigma_j^i < s} g_i(s - \sigma_j^i).$$

As a function of g we note that $X_s : \mathcal{H}^p \to \mathbb{R}$ being a sum of function evaluations is a continuous linear functional. The process $X_s(g)$ is called the *linear predictor* process. We consider the model of N where the intensity is given as $\lambda_s(g) = \varphi(X_s(g))$ with $\varphi : \mathbb{R} \to [0, \infty)$ a known function, and the objective is to estimate the g_i-functions in $\mathcal{H}$.

From Corollary II.7.3 in [1] it follows that the negative log-likelihood w.r.t. the homogeneous Poisson process is given as

$$\ell(g) = \int_0^t \varphi(X_s(g)) \, \mathrm{d}s - \sum_{k=1}^{N_t} \log \varphi(X_{\tau_k}(g)). \tag{3}$$

If φ is the identity the time integral has a closed form representation in terms of the antiderivatives of g_i, but in general it has to be computed numerically.

The following proposition gives the gradient of ℓ in the reproducing kernel Hilbert space. This result is central for our development and understanding of a practically implementable minimization algorithm of the penalized negative log-likelihood.

Proposition 1 *If φ is continuously differentiable the gradient in $\mathcal{H}$ w.r.t. g_i is*

$$\nabla_i \ell(g) = \sum_j \int_{\sigma_j^i}^t \varphi'(X_s(g)) R(s - \sigma_j^i, \cdot) \, \mathrm{d}s - \sum_j \sum_{k : \sigma_j^i < \tau_k} \frac{\varphi'(X_{\tau_k}(g))}{\varphi(X_{\tau_k}(g))} R(\tau_k - \sigma_j^i, \cdot). \tag{4}$$

The proof of Proposition 1 is given in Section 6. It is a special case of Proposition 3.6 in [7] if $\mathcal{H}$ is a Sobolev space. However, since we restrict attention to counting process integrators in this paper, in contrast to [7] where more general integrator processes are allowed, we can give a relatively elementary proof for $\mathcal{H}$ being any reproducing kernel Hilbert space with a continuous kernel.

Computations of ℓ as well as the gradient involve the computation of $X_s(g)$. Without further assumptions a direct computation of $X_s(g)$ on a grid of n time points involves in the order of $n \sum_{i=1}^p N_t^i$ evaluations of the

g_i-functions. In comparison, (1) can be computed recursively with the order of np evaluations of the exponential function.

In this paper we consider three techniques for reducing the general costs of computing $X_s(g)$.

- **Bounded memory**. The filter functions g_i are restricted to have support in $[0, A]$ for a fixed A.
- **Preevaluations**. The filter functions are preevaluated on a grid in $[0, A]$.
- **Basis expansions**. The filter functions are of the form $g = \sum_k \beta_k B_k$ for fixed basis functions B_k and

$$X_s(g) = \sum_k \beta_k X_s(B_k).$$

The linear filters $X_s(B_k)$ are precomputed.

3 Time discretization

In this section we discuss the time discretizations necessary for the practical implementation of an optimization algorithm in $\mathcal{H}$. We assume that all filter functions g_i have a prespecified support restricted to $[0, A]$, and that $\mathcal{H}$ is restricted to be a space of functions with support in $[0, A]$. We approximate time integrals by left Riemann sums with functions evaluated in the grid

$$0 = t_0 < t_1 < \ldots < t_n = t$$

and corresponding interdistances $\Delta_l = t_l - t_{l-1}$ for $l = 1, \ldots, n$. We will assume that the collection of jump times for N^i for $i = 1, \ldots, p$ is a subset of this grid and denote the corresponding subset of indices by $I_{\text{jump}} \subseteq \{0, \ldots, n\}$.

We need an implementable representation of the linear predictor as well as the functional gradient. A possible representation of g_i itself is via the N-dimensional vector $\mathbf{g}_i$ of its evaluations in a grid

$$0 = \delta_0 < \delta_1 < \ldots < \delta_N = A,$$

that is, $\mathbf{g}_{ik} = g_i(\delta_k)$ for $k = 0, \ldots, N-1$. We let $\mathbf{g}$ denote the $N \times p$ matrix with columns $\mathbf{g}_i$'s for $i = 1, \ldots, p$. Define

$$h_{lik} = \#\{j \mid \delta_k \leq t_l - \sigma_j^i < \delta_{k+1}\} 1(\sigma_j^i < t_l)$$

as the number of jumps for N^i in $(t_l - \delta_{k+1}, t_l - \delta_k]$. The indicator $1(\sigma_j^i < t_l)$ ensures that if $t_l = \sigma_j^i$ then $h_{li0} = 0$, which, in turn, ensures that the approximation of the linear predictor below does not anticipate jumps. It is the intention that the grids are chosen such that the h_{lik}'s take the values 0 and 1 only. The linear predictor for given g_i's evaluated in the grid points is approximated as

$$\begin{aligned}\xi_l &:= \sum_{i,k} h_{lik} \mathbf{g}_{ik} \qquad (5)\\ &\simeq \sum_{j : t_l - A \leq \sigma_j^i < t_l} g(t_l - \sigma_j) = \int_{t_l - A}^{t_l -} g(t_l - u) \mathrm{d}N_u^i.\end{aligned}$$

Strictly speaking, for correct handling of the lower limit in the integral, $h_{li(N-1)}$ should be redefined to be 1 if $\sigma_j^i = t_l - A$. Such a redefinition will typically have no detectable consequences, whereas handling the case $\sigma_j^i = t_l$ correctly is crucial to avoid making the approximation anticipating. An approximation of the negative log-likelihood in g is then obtained as

$$\ell^{\text{approx}}(\mathbf{g}) = \sum_l \varphi(\xi_l) \Delta_l - \sum_{l \in I_{\text{jump}}} \log \varphi(\xi_l). \qquad (6)$$

If we use the same δ-grid for evaluating the kernel R we get the gradient approximation from Proposition 1

$$\begin{aligned}\nabla_i \ell^{\text{approx}}(\mathbf{g}) = &\sum_k \left(\sum_l \varphi'(\xi_l) \Delta_l h_{lik} \right) R(\delta_k, \cdot)\\ &- \sum_k \left(\sum_{l \in I_{\text{jump}}} \frac{\varphi'(\xi_l)}{\varphi(\xi_l)} h_{lik} \right) R(\delta_k, \cdot). \qquad (7)\end{aligned}$$

We observe that

$$\nabla_i \ell^{\text{approx}}(\mathbf{g}) \in \text{span}\{R(\delta_0, \cdot), \ldots, R(\delta_{N-1}, \cdot)\}.$$

The consequence is that any descent algorithm based on $\nabla_i \ell^{\text{approx}}(\mathbf{g})$ stays in the finite dimensional subspace spanned by $R(\delta_0, \cdot), \ldots, R(\delta_{N-1}, \cdot)$ – if we start in this subspace. As we show below, there is a unique element in this subspace with evaluations $\mathbf{g}_i$, and the discretization effectively restricts g_i to be a function in this subspace.

3.1 Method A – the direct approach

The $N \times N$ Gram matrix $\mathbb{G}$ is given as $\mathbb{G}_{kl} = R(\delta_k, \delta_l)$. The vector $\mathbf{g}_i$ can be identified with the unique function $g_i = \sum_k \beta_{ik}^0 R(\delta_k, \cdot)$ obtained by solving

$$\mathbf{g}_i = \mathbb{G} \beta_i^0.$$

This is the minimal norm element whose evaluations coincide with $\mathbf{g}_i$. Since the Gram matrix is positive definite, $\mathbb{G} = UU^T$ where the columns of U are orthogonal. This gives that

$$\mathbf{g}_i = U \underbrace{U^T \beta_i^0}_{\beta_i} = U \beta_i.$$

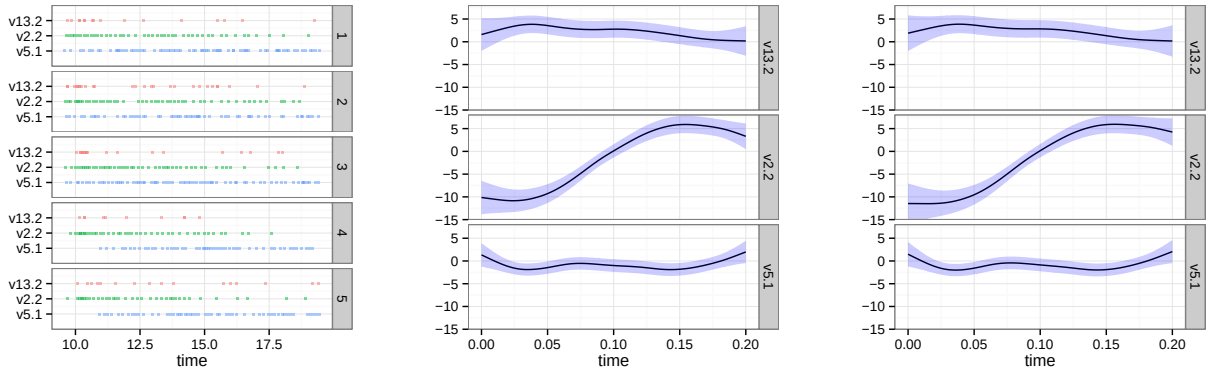

Fig. 1 Left: Data example consisting of three spike tracks from five independent multichannel measurements of turtle spinal neurons during a stimulation period. Middel: Estimates of the h_i's in the model of v2.2 using method A to minimize (8) with $n = 7752$ and $N = 120$. The value of $\lambda = 1.2 \times 10^{-4}$ was chosen data adaptively. Right: Similar estimates of the h_i's using method B to minimize (10) with $q = 50$, with the basis functions being B-splines and with $\lambda = 9.5 \times 10^{-7}$ chosen data adaptively. The point-wise approximative 95% confidence intervals were obtained using a sandwich estimator of the asymptotic parameter variance.

Note how the β_i^0- and thus the β_i-parameter representation of the evaluations $\nabla_i \ell^{\text{approx}}(\mathbf{g})(\delta_k)$ can be read of directly from (7). We observe that the squared norm of g_i equals

$$||g_i||^2 = (\beta_i^0)^T \mathbb{G} \beta_i^0 = ||\beta_i||_2^2$$

with $||\cdot||_2$ denoting the ordinary Euclidean norm on $\mathbb{R}^N$. The parametrization in terms of β_i is thus an isometry from $\mathbb{R}^N$ into $\mathcal{H}$. The objective function – the penalized negative log-likelihood approximation – can be computed as

$$\ell^{\text{approx}}(U\beta) + \lambda \sum_i ||\beta_i||_2^2 \tag{8}$$

using (6) and the β_i-gradient can be computed as

$$U^T \nabla_i^\beta \ell^{\text{approx}}(U\beta) + 2\lambda\beta_i$$

where

$$\nabla_i^\beta \ell^{\text{approx}}(\mathbf{g})_k = \sum_l \varphi'(\xi_l) \Delta_l h_{lik} - \sum_{l \in I_{\text{jump}}} \frac{\varphi'(\xi_l)}{\varphi(\xi_l)} h_{lik} \tag{9}$$

Using (6) and (9) – and (5) – requires the computation of h_{lik}. This can either be done on-the-fly (a matrix free method) or by precomputing the $n \times (pr)$-dimensional sparse matrix $\mathbf{H} = (h_{lik})$.

3.2 Method B – using basis expansions

Choose a set of basis functions $B_1, \ldots, B_q$ such that

$$\text{span}\{B_1, \ldots, B_q\} \subseteq \text{span}\{R(\delta_0, \cdot), \ldots, R(\delta_{N-1}, \cdot)\}.$$

Precompute the $n \times q$ *model matrices* $\mathbf{Z}^i$ of basis filters

$$\mathbf{Z}^i_{lj} = \sum_k h_{lik} B_j(\delta_k).$$

With $g_i = \sum_j \beta_{ij}^0 B_j$, the n-dimensional linear predictor is given as $\xi = \sum_i \mathbf{Z}^i \beta_i^0$ and $\ell^{\text{approx}}(\beta^0)$ can be computed using (6). The Gram matrix, $\mathbb{G}$, is given by $\mathbb{G}_{kl} = \langle B_k, B_l \rangle$. It is positive definite, thus with $V \mathbb{G} V^T = I$ for V a matrix with orthogonal columns we reparametrize in terms of $\beta_i = V^{-1}\beta_i^0$. In this parametrization

$$||g_i||^2 = (\beta_i^0)^T \mathbb{G} \beta_i^0 = ||\beta_i||_2^2,$$

thus β_i provides an isometric parametrization from $\mathbb{R}^q$ into $\mathcal{H}$. The objective function becomes

$$\ell^{\text{approx}}(V\beta) + \lambda \sum_i ||\beta_i||_2^2 \tag{10}$$

and the gradient is

$$\begin{aligned}
&\sum_l \varphi'(\xi_l) \Delta_l (\mathbf{Z}_l^i V)^T - \sum_{l \in I_{\text{jump}}} \frac{\varphi'(\xi_l)}{\varphi(\xi_l)} (\mathbf{Z}_l^i V)^T + 2\lambda\beta_i \\
&= V^T \left(\sum_l \varphi'(\xi_l) \Delta_l (\mathbf{Z}_l^i)^T - \sum_{l \in I_{\text{jump}}} \frac{\varphi'(\xi_l)}{\varphi(\xi_l)} (\mathbf{Z}_l^i)^T \right) + 2\lambda\beta_i \\
&= V^T \nabla_i^0 l_t^{\text{approx}}(V\beta) + 2\lambda\beta_i
\end{aligned}$$

where

$$\nabla_i^0 l_t^{\text{approx}}(\beta^0) = \sum_l \varphi'(\xi_l) \Delta_l (\mathbf{Z}_l^i)^T - \sum_{l \in I_{\text{jump}}} \frac{\varphi'(\xi_l)}{\varphi(\xi_l)} (\mathbf{Z}_l^i)^T$$

is the gradient in the β_i^0 parametrization.

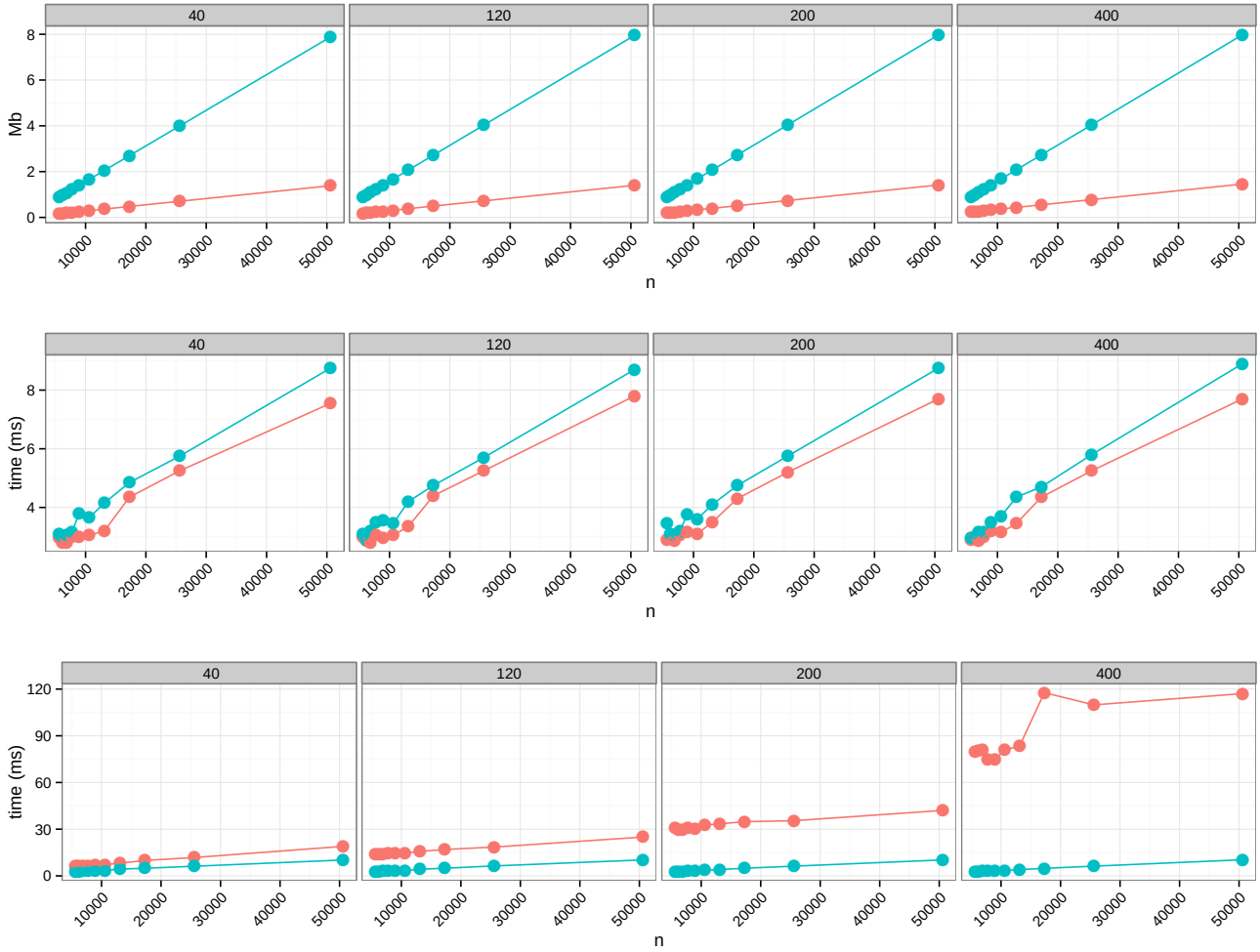

Fig. 2 Top: Memory usage for storing the **H**-matrix for method A (•) and the **Z**-matrix for method B (•) for N ranging from 40 to 400. Middel: Computation time for computing the negative log-likelihood. Bottom: Computation time for computing the gradient.

4 Results

We have implemented Method A and Method B and applied them to a test data set of neuron spike times. The data set consisted of multichannel measurements of spinal neurons from a turtle. The measurements were replicated 5 times and each time the spike activity was recorded over a period of 40 seconds. A 10 seconds stimulation was given within the observation window. We used the spike times for 3 neurons during the stimulation period, see Figure 1

The likelihood and gradient algorithms are implemented in the R-package `ppstat`, which supports optimization of the objective functions via the R-function `optim` using the BFGS-algorithm. The `ppstat` package offers a formula based model specification with an interface familiar from `glm`. Method A is implemented via the `ppKernel` function, with a typical call of the form

```
ppKernel(v2.2 ~ k(v13.2) + k(v2.2) + k(v5.1),
                    data = spikeData,
                    family = Hawkes("logaffine"),
                    support = 0.2,
                    N = 120
)
```

The data set contained in the object `spikeData` must be of class `MarkedPointProcess` from the supporting R-package `processdata`. Method B is implemented via the similar `ppSmooth` function. The choice of φ is specified by the "inverse link function" – being `"logaffine"` in the call above. This function will be used throughout, and it is given as

$$\varphi(x) = \begin{cases} e^x & \text{for } x \leq 0 \\ x + 1 & \text{for } x > 0. \end{cases}$$

It maps $\mathbb{R}$ into $(0, \infty)$ and is continuously differentiable. The benefit of using this φ over the exponential function is that the exponential function often produces models

that are unstable or even explodes in finite time. We will not pursue the details. See [5] for details on stability.

Figure 1 shows the estimated h_i's obtained using either method A with the Sobolev kernel or method B with a B-spline basis. The estimates were computed by minimizing (8) and (10), respectively. The choice of the penalization parameter was made data adaptively by minimizing a TIC-criterion, see [6]. We will not pursue the details of the model selection procedure here, but focus on the efficiency of the computations of the likelihood and gradient.

We investigated the memory usage and the computation times of both methods. The memory usage was obtained using the R-function `object.size` and the computation times were computed as the average of 40 replicated likelihood or gradient evaluations. The interest was on how they scale with the numbers n and N that determine the resolution of the time discretization. The number of basis functions for method B was kept fixed and equal to $q = 50$. Due to this fact N only affects the precomputation of the model matrices and not the likelihood and gradient computations for Method B. The implementation relies on precomputation of the $\mathbf{H}$ or $\mathbf{Z}$ matrices, which are stored as sparse matrices as implemented in the R-package `Matrix`.

Figure 2 shows that Method B used more memory for storing $\mathbf{Z}$ and that the memory usage as a function of n had a somewhat larger slope for Method B than for Method A. We should also note that the memory usage for Method A does not depend upon N. Storing the matrices as non-sparse matrices the $\mathbf{Z}$-matrix required 61 MB and the $\mathbf{H}$-matrix required 464 MB for $n = 50000$ and $N = 400$. In comparison, the sparse versions required 8 MB and 1.5 MB, respectively.

Figure 2 shows, furthermore, that the methods were comparable in terms of the time to compute the likelihood – with Method A being slightly faster. There were again no detectable dependence on N for Method A. For the gradient computations, Figure 2 shows that Method A was slowest and scaled badly with N.

5 Discussion

The two methods, Method A and Method B, considered in this paper differ in terms of what is precomputed. Computing the matrix $\mathbf{H}$ upfront (or computing it on-the-fly) as in Method A should require only a fraction of the memory for storing the $\mathbf{Z}$-matrices as most entries are 0. This was confirmed by our implementation. We also confirmed that neither the storage requirements nor the likelihood computations for Method A depend on the number N of δ-grid points when $\mathbf{H}$ is stored as a sparse matrix. However, the gradient computations for Method A were clearly dominated by the N^2 scaling.

The storage requirements for $\mathbf{Z}$ can easily become prohibitively large. A choice of basis functions with local support in Method B, such as B-splines used here, can compensate partly for this. It is unlikely that it is useful to precompute $\mathbf{Z}^i V$, as this will destroy the benefits of a local basis.

For Method B it is possible to precompute the model matrix in a sligthly different and more direct way. Instead of precomputing the $q \times N$ basis function evaluations $B_j(\delta_k)$ we can compute $\mathbf{Z}^i_{lj}$ directly as

$$\mathbf{Z}^i_{lj} = \sum_{k : t_l - A < \sigma^i_k < t_i} B_j(t_l - \sigma^i_k).$$

This may be more accurate but since $n \gg N$ in typical applications this comes at the cost of many more basis function evaluations. Whether this is critical in terms of the time to compute $\mathbf{Z}^i$ depends upon how costly a single basis function evaluation is relative to the computation of the h_{lik}'s. We have not presented data on the computational costs of the precomputations, but they were observed to be small compared to the costs of the actual optimization.

6 Proof of Proposition 1

First note that since function evaluations are represented in terms of the kernel by inner products we have that

$$\begin{aligned} X_s(g) &= \sum_{i=1}^{p} \sum_{j : \sigma^i_j < s} \langle R(s - \sigma^i_j, \cdot),\ g_i \rangle \\ &= \sum_{i=1}^{p} \left\langle \sum_{j : \sigma^i_j < s} R(s - \sigma^i_j, \cdot),\ g_i \right\rangle . \end{aligned} \tag{11}$$

If ψ is a continuous differentiable function we find that

$$\begin{aligned} &\frac{\psi(X_s(g + \varepsilon h)) - \psi(X_s(g))}{\varepsilon} \\ &= \frac{\psi(X_s(g) + \varepsilon X_s(h)) - \psi(X_s(g))}{\varepsilon} \longrightarrow \psi'(X_s(g)) X_s(h) \end{aligned}$$

for $\varepsilon \to 0$. This is clearly a continuous linear functional. Using (11) and differentiating only w.r.t. the i'th coordinate of g we find that the corresponding gradient in $\mathcal{H}$ is

$$\nabla_i \psi(X_s(g)) = \psi'(X_s(g)) \sum_{j : \sigma^i_j < s} R(s - \sigma^i_j, \cdot).$$

Taking $\psi = \log \varphi$ this yields the gradient of the second term in the negative log-likelihood, $\sum_{k=1}^{N_t} \log \varphi(X_{\tau_k}(g))$,

directly. For the first term we take $\psi = \varphi$, but we need to ensure that we can interchange the order of integration and differentiation. To this end the following norm bound on $\nabla_i \varphi(X_s(g))$ is useful

$$||\nabla_i \varphi(X_s(g))|| \leq |\varphi'(X_s(g))| \sum_{j:\sigma_j^i < s} ||R(s - \sigma_j^i, \cdot)||$$
$$\leq C_t N_t^i \sup_{s \in [0,t]} \sqrt{R(s,s)} < \infty.$$

Here $C_t = \sup_{s\in[0,t]} |\varphi'(X_s(g))|$ is finite because $X_s(g)$ is continuous in s and φ' is assumed continuous. We have also used that $||R(s - \sigma_j^i, \cdot)||^2 = R(s - \sigma_j^i, s - \sigma_j^i)$ and the fact that R is continuous to conclude that the bound is finite. The bound shows that

$$\sum_j \int_{\sigma_j^i}^t \varphi'(X_s(g)) R(s - \sigma_j^i, \cdot)\, \mathrm{d}s$$

is an element in $\mathcal{H}$ and the required interchange of integration and differentiation is justified by the bound. This completes the proof. □

Acknowledgements The neuron spike data were provided by Associate Professor, Rune W. Berg, Department of Neuroscience and pharmacology, University of Copenhagen.

References

1. Andersen, P.K., Borgan, Ø., Gill, R.D., Keiding, N.: Statistical models based on counting processes. Springer Series in Statistics. Springer-Verlag, New York (1993)
2. Anderson, D., Kurtz, T.: Continuous time markov chain models for chemical reaction networks. In: H. Koeppl, G. Setti, M. di Bernardo, D. Densmore (eds.) Design and Analysis of Biomolecular Circuits, pp. 3–42. Springer New York (2011). DOI 10.1007/978-1-4419-6766-4_1. URL `http://dx.doi.org/10.1007/978-1-4419-6766-4_1`
3. Berlinet, A., Thomas-Agnan, C.: Reproducing kernel Hilbert spaces in probability and statistics. Kluwer Academic Publishers, Boston, MA (2004). With a preface by Persi Diaconis
4. Bowsher, C.G.: Stochastic kinetic models: Dynamic independence, modularity and graphs. Annals of Statistics **38**(4), 2242–2281 (2010)
5. Brémaud, P., Massoulié, L.: Stability of nonlinear Hawkes processes. Ann. Probab. **24**(3), 1563–1588 (1996)
6. Burnham, K.P., Anderson, D.R.: Model selection and multimodel inference, second edn. Springer-Verlag, New York (2002). A practical information-theoretic approach
7. Hansen, N.R.: Penalized maximum likelihood estimation for generalized linear point processes pp. 1–34 (2013). URL `http://arxiv.org/abs/1003.0848`
8. Hautsch, N.: Modelling irregularly spaced financial data, *Lecture Notes in Economics and Mathematical Systems*, vol. 539. Springer-Verlag, Berlin (2004). Theory and practice of dynamic duration models, Dissertation, University of Konstanz, Konstanz, 2003
9. Hawkes, A.G.: Spectra of some self-exciting and mutually exciting point processes. Biometrika **58**(1), pp. 83–90 (1971). URL `http://www.jstor.org/stable/2334319`
10. Jacobsen, M.: Point process theory and applications. Probability and its Applications. Birkhäuser Boston Inc., Boston, MA (2006). Marked point and piecewise deterministic processes
11. Pillow, J.W., Shlens, J., Paninski, L., Sher, A., Litke, A.M., Chichilnisky, E.J., Simoncelli, E.P.: Spatio-temporal correlations and visual signalling in a complete neuronal population. Nature **454**, 995–999 (2008)